

Université François-Rabelais de Tours
Institut Universitaire de Technologie de Tours
Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS
TOURS



Institut Universitaire de Technologie

Département
GENIE ELECTRIQUE ET
INFORMATIQUE INDUSTRIELLE



Vu-mètre à LED pour une batterie de voiture 12V

Arsen ARAKELIAN
Clément GALLIER
K4 A
Promo 2010-2013

Enseignants
Thierry. LEQUEU
Philippe AUGER

Remerciement

Tout d'abord, nous remercions notre professeur M.LEQUEU qui nous a aidé tout au long de notre projet et de l'aide que son site nous a apporté. Nous remercions aussi Richard le magasinier qui a passé nos commandes et enfin nous remercions M.AUGER pour avoir porté attention à notre projet.

Sommaire

Remerciement	3
Introduction.....	5
1.Cahier des charges.....	7
2.Le Planning.....	7
3.Analyse générale du Vu-Mètre.....	8
4.Analyse technique.....	10
4.1.Alimentation à découpage.....	10
4.2.Réduction de la tension.....	12
4.3.Contrôle des LED suivant la tension d'entrée.....	13
5.Réalisation.....	14
5.1.Réalisation de la carte 1.....	14
5.2.Réalisation de la carte 2.....	28
5.3.Problèmes rencontrés.....	31
5.4.Nomenclature.....	32
Conclusion.....	33
Résumé.....	34
Annexe 1 : Schéma de construction de la boîte pour la carte 2.....	38
Annexe 2 : Schéma du 2ème ATmega8535.....	39

Introduction

Le vu-mètre que nous allons fabriquer va nous permettre de visualiser la tension que nous avons sur notre batterie de 12V. Ce projet pourra être très utile pour les vieilles voitures ou celles qui ont des problèmes de batterie. Pour réaliser ce projet, nous allons étudier les montages qui nous permettront d'alimenter un micro-contrôleur pour que celui-ci vérifie la tension de notre batterie de voiture.

Dans ce dossier, vous trouverez les explications des solutions technologiques utilisées pour visualiser la tension de notre batterie de voiture. En premier lieu, le cahier des charges sera défini. Dans un deuxième temps, un planning de réalisation de ce projet sera mis en place. Puis, une analyse générale de ce système sera proposée pour suivre sur une technique qui présentera les différentes parties du projet. Pour finir, les réalisations effectuées vous seront présentées.

1. Cahier des charges

Présentation du projet

Nous allons réaliser un vu-mètre à LED pour vérifier la tension de charge d'une batterie 12V. Nous le brancherons sur l'allume cigare de la voiture. Si la batterie est chargée toutes les LED seront allumées et si elle n'est pas chargée les LED seront éteinte. Chaque LED affichera un seuil de tension différent. Les différents seuils de tension seront vérifiés à l'aide d'un ATmega8535.

Les contraintes

- L'utilisation d'un circuit intégré ATmega8535.
- L'utilisation d'une batterie au plomb 14,6V max.
- Une alimentation à découpage de 5V pour alimenter l'ATmega8535.
- Prise allume cigare.

Les composants

- Un micro-contrôleur ATmega8535.
- Adaptateur allume cigare de voiture.
- Alimentation à découpage.
- Pont diviseur de tension.

Description du système

- Recevoir la tension de la batterie.
- Alimenter le micro-contrôleur.
- Alimenter plus ou moins de LED avec le micro contrôleur suivant la tension de la batterie.

2. Le Planning

Nous avons réalisé un planning prévisionnel (ici en bleu) et notre planning réel a été ajouté dans le même tableau. Nous pouvons donc constater que le planning réel est très proche du planning prévisionnel jusqu'à la semaine 40.

Taches	Semaines										
	37	38	39	40	41	42	43	44	45		
Étude du projet	Red	Red								Green	
Analyse et développement		Red	Red							Green	
Schémas électrique et typon			Red	Red	Red		Blue			Green	
Gravure, Soudure et Teste				Red	Red	Red		Blue	Blue	Green	
Partie programmation de l'ATmega				Blue	Blue		Red	Red	Red	Green	
Validation des testes								Red	Red	Green	
Rédaction du rapport									Red	Red	
Oral										Green	Red
Planning Prévisionnelle	Red										
Préparation de l'exposer oral et Vacances	Green										
Planning Réelle	Blue										

Illustration 1: Planning.

Nous avons fini l'analyse et le développement du projet plus tôt que prévu (semaine 37), donc nous avons commencé le schéma structurel et le typon de notre projet. Nous avons pu commencer la programmation plus tôt, à partir de la semaine 40 le retour en arrière correspond au début de notre 2^{ème} carte. Cela explique que la suite du planning réel ne correspond plus au planning prévisionnel.

3. Analyse générale du Vu-Mètre.

Schéma fonctionnel de niveau 1

Le vu-mètre que nous allons fabriquer va nous permettre de voir si la batterie de notre voiture est chargée. Les LED¹ nous serviront à afficher la tension de la batterie (indicateur de charge).

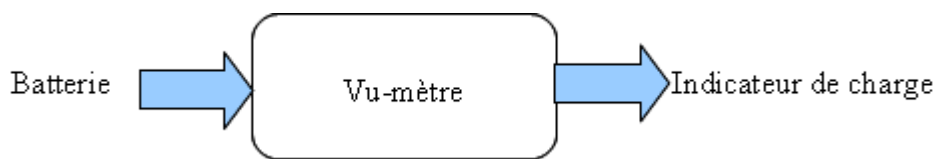


Illustration 2: Schéma fonctionnel de niveau 1 du projet. [3]

Lorsqu'il y a au moins une LED d'allumée la tension de la batterie sera d'au moins 10V. Le vu-mètre permet de mesurer la valeur entre 10 et 14,5V. Chaque LED sera allumée à un seuil de tension différent.

Présentation des différents seuils de tension des diodes

Nombres de diodes du VU-Mètre	Seuil de tension de batterie pour chaque diode
LED 1 (Verte)	14,5V
LED 2 (Verte)	14,0V
LED 3 (Verte)	13,6V
LED 4 (Jaune)	13,0V
LED 5 (Jaune)	12,0V
LED 6 (Rouge)	11,5V
LED 7 (Rouge)	11,0V
LED 8 (Rouge)	10,0V

Chaque LED restera allumée tant que son seuil de tension ne sera pas dépassé. Donc toutes les LED seront allumés lorsque la tension sera de 14,5V.

Schéma fonctionnel de niveau 2

Ce schéma fonctionnel représente le vu-mètre. C'est le niveau supérieur du schéma fonctionnel précédent. Donc un niveau plus détaillé de la fonction principale.

¹ LED (light-emitting diode), c'est des diodes électroluminescente.

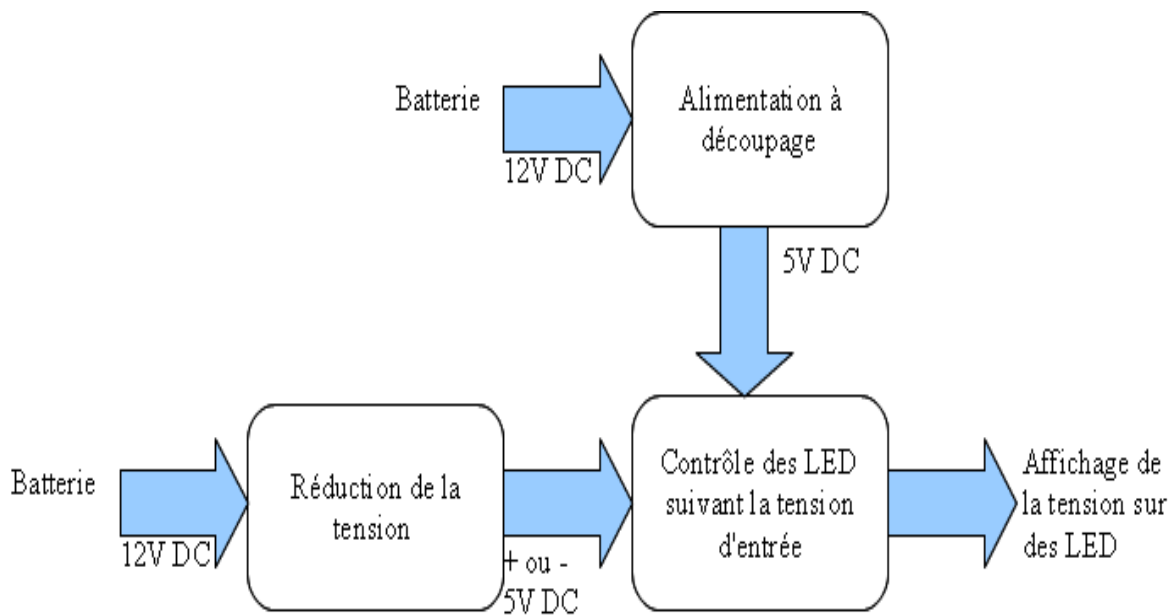


Illustration 3: Schéma fonctionnel de niveau 2 du projet. [3]

La batterie sera d'environ 12V, elle aura un maximum de 14,5V. Sa variation de tension est due à sa charge ou à sa décharge.

Le pont diviseur de tension va permettre au micro-contrôleur de visualiser la tension de la batterie sans pour autant détériorer celui-ci.

Elle va permettre d'alimenter en 5V DC² le micro-contrôleur pour que celui-ci puisse mesurer la tension de la batterie.

Nous pouvons voir que le micro-contrôleur va permettre d'alimenter plus ou moins de LED en fonction de la tension mesurée.

4. Analyse technique

4.1. Alimentation à découpage

Pour l'alimentation du micro contrôleur nous allons utiliser une alimentation à découpage.

Schéma fonctionnel de niveau 1

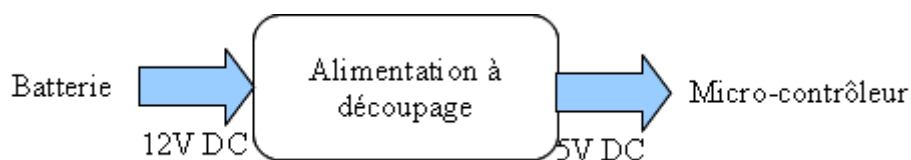


Illustration 4: Schéma fonctionnel de niveau 1 de l'alimentation à découpage. [3]

² 5V DC signifie 5V courant continu.

La réduction de la tension continue de 12 V à 5 V a été vue en première année de ce DUT lors de l'étude de la carte d'alimentation au cours du semestre 2. Nous avons repris ce montage lors de la séance 2 en recalculant les valeurs des condensateurs et de l'inductance.

Voici le montage d'un réducteur de tension

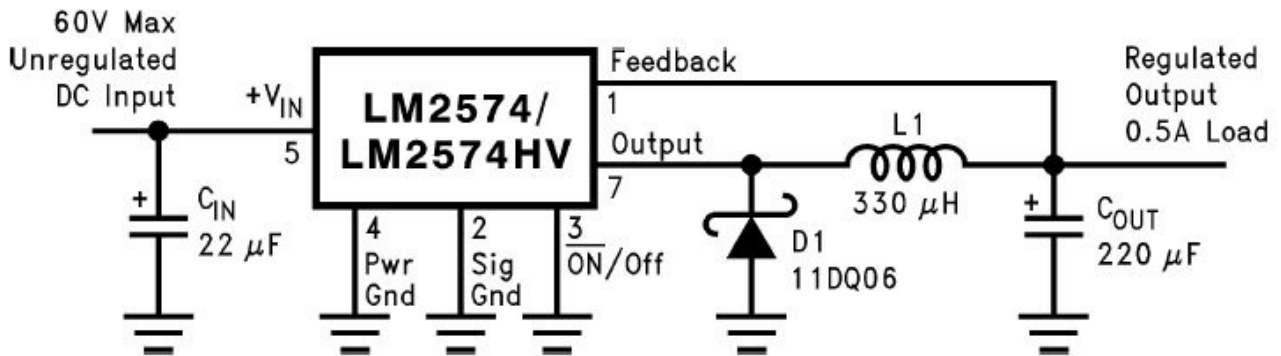


Illustration 5: Schéma structurel d'un réducteur de tension. [1]

Ce montage correspond à une alimentation à découpage.

Il est écrit sur le schéma qu'il faut une tension maximale de 60 V continu en entrée. Nous avons une tension de 12 V continu en entrée, ce qui est bien inférieur à 60 V et elle est bien continu. Le condensateur d'entrée que nous utilisons est un $C_{IN} = 470\mu\text{F}$ (valeur calculé au semestre 2).

C'est le LM2574³ qui crée le découpage, ses pattes 2 et 4 ne sont pas utilisées donc elles seront reliées à la masse (Gnd). La patte 3 est aussi reliée à la masse, car celle-ci permet la mise en marche ou arrêt du régulateur. Donc pour que celui-ci fonctionne en permanence, il faut que la patte soit à la masse. La patte 7 est la sortie (Output) du LM2574. La bobine (L1) présente en sortie va permettre de stocker une énergie magnétique, ce qui va rendre possible le fonctionnement en continu malgré la tension découpée obtenue en sortie du LM2574, car elle va restituer cette énergie. Le condensateur de sortie (Cout) est utilisé pour la continuité de la tension. La valeur de ce condensateur ainsi que celle de la bobine ont été reprises sur le montage de notre professeur M. LEQUEU [1], car il avait ce même montage avec des valeurs de tension et courant de sorties identiques. La diode schottky (D1) ferme la boucle pour laisser la circulation du courant et de la tension libre. Nous utilisons une diode schottky, car celle-ci à une chute de tension très petite ce qui est recherché pour avoir une tension continue. De plus, le temps de réponse est très petit ce qui est souhaité, car nous sommes en 50 kHz. Le feedback sert à réguler la tension de sortie.

3 LM2574 : c'est un régulateur de tension (12 V à 5 V DC).

4.2. Réduction de la tension

Schéma fonctionnel de niveau 1

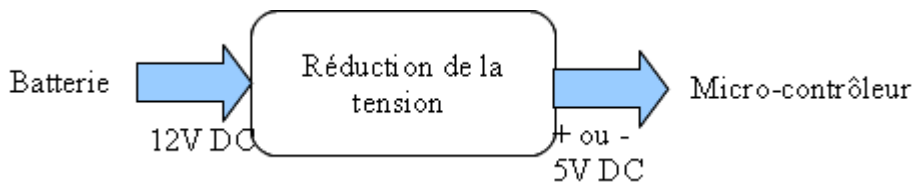


Illustration 6: Schéma fonctionnel de niveau 1 de la réduction de tension. [3]

Pour avoir 5V nous utiliserons un pont diviseur de tension. Ce qui nous permettra de réduire la tension de 12V à 5V. Un pont diviseur de tension est constitué de deux résistances.

Comme nous pouvons le voir, la tension d'entrée V_e correspond sur notre projet à la tension de batterie et la tension de sortie V_s alimente le micro-contrôleur. Pour calculer les deux résistances nous utiliserons la loi du pont diviseur de tension.

Calcul des résistances R1 et R2

La résistance R1 correspond à la résistance R10 et R2 à R9 sur le schéma principal de notre projet.

Pour pouvoir calculer R2 on fixe R1 à 4,7k Ω .

Notre tension de sortie est de 5V et la tension d'entrée maximal est de 14,6V.

$$V_s = V_e \times \left(\frac{R1}{R1 + R2} \right)$$

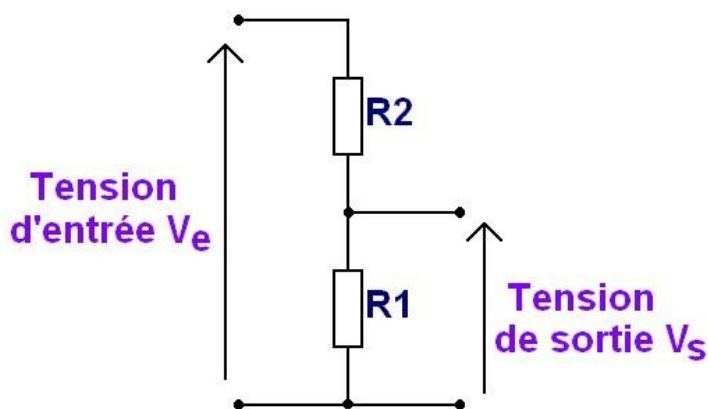


Illustration 7: Pont diviseur de tension. [2]

$$R2 = \frac{V_e \times R1}{V_s} - R1 = \frac{14,6 \times 4,7 \times 10^3}{5} - 4,7 \times 10^3 = 9,02 \text{ k}\Omega$$

La valeur de la résistance normaliser est R2=9,1k Ω .

4.3. Contrôle des LED suivant la tension d'entrée

Dans cette partie, nous allons expliquer comment réaliser la fonction souhaitée.

Schéma fonctionnel de niveau 1

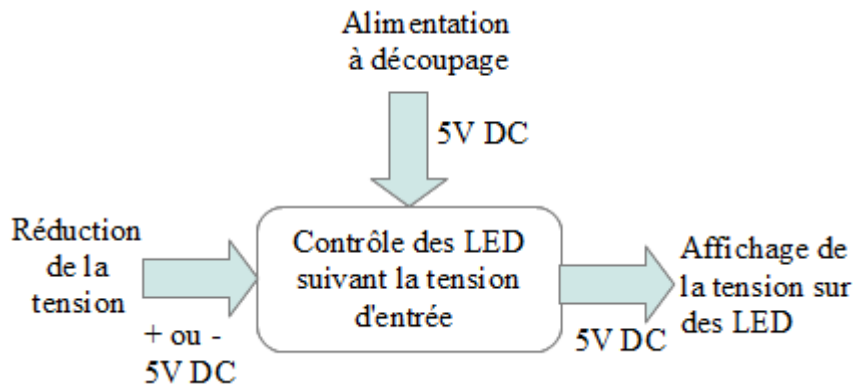


Illustration 8: Schéma fonctionnel du Contrôle des LED.[3]

4.3.1. Micro-contrôleur

Le micro-contrôleur que nous utilisons est un ATmega8535, car c'est le plus connu à l'IUT. Nous utiliserons cet ATmega8535⁴, car le logiciel de programmation est déjà installé sur les ordinateurs de l'IUT et sa programmation n'est pas très compliquée.

Le logiciel que nous avons utilisé est code vision AVR.



Illustration 9: Icône de code vision AVR.

Grâce au programme le micro-contrôleur pourra vérifier la tension de la batterie. En fonction de la tension mesurée le micro-contrôleur allumera plus ou moins de LED.

Les différents ATmega8535 utilisés

Nous avons utilisé deux ATmega8535, le premier présenté dans l'illustration 10 est un ATmega8535 de 40 broches qui a un trop grand boîtier pour être utilisé dans notre carte finale qui sera plus petite que la carte de test. Donc l'ATmega8535 que nous utiliserions sur notre carte finale est présenté sur l'illustration 11. Les deux ATmega8535 ont un fonctionnement identique, seuls leur boîtiers changent. Donc dans le 44 broches il sort plus de masse et de plus Vcc⁵.

4 ATmega8535 : c'est le composant programmable qui permet de gérer la commande de coupure de l'alimentation (tension de 5 V DC).

5 Vcc est la tension d'alimentation principale de l'ATmega.



Illustration 10: ATmega8535 en longueur de 40 pattes. [1]



Illustration 11: ATmega8535 en carré de 44 pattes.

4.3.2. Utilisation des LED

Les LED sont alimentées par le micro-contrôleur en 5V, les LED que nous utilisons doivent avoir un courant d'environ 2mA. Pour cela nous devons mettre des résistances après les LED. Les LED ont un fonctionnement identique à celui d'une diode normale, elles laissent passer le courant que dans un seul sens.

Calcul des résistances [3]

Nous savons que la tension en sortie du micro-contrôleur est de 5V DC, le courant dans la LED doit être de 2mA. Pour calculer la résistance nous utiliserons la loi d'ohm⁶.

$U = 5V$ (Tension de sortie de l'ATmega8535).

$I = 2mA$ (Courant dans la LED donné par le fabricant).

$U = R \times I$ Donc pour trouver R,

$$R = \frac{U}{I} = \frac{5}{2 \times 10^{-3}} = 2,5 \text{ k}\Omega$$

Nous prenons la valeur normalisée de 2,7k Ω .

Derrière chaque LED nous avons une résistance de 2,7k Ω .

5. Réalisation

5.1. Réalisation de la carte 1

5.1.1. Schéma structurel

Lors de la première séance de réalisation nous avons créé le schéma structurel sur le logiciel ORCAD. Nous avons inséré les différents composants de notre montage et relié les différentes broches.

6 Loi d'ohm : $U = R \times I$

Durant la troisième séance, le schéma structurel du projet a été réuni, car le schéma de l'ATmega8535 a pu être récupéré auprès de notre professeur, sur son site. Nous l'avons adapté et modifié à notre projet. Le schéma structurel a été séparé en trois parties (vu dans l'analyse technique) voir ci-dessous.

Au centre du schéma structurel nous avons la partie ATmega8535 (U1) [1], le connecteur JP1 nous permet de programmer le micro-contrôleur. Les broches du port de sortie C sont reliés aux différentes LED (en haut du schéma structurel). La tension de la batterie arrive sur le port A0, en passant par le pont diviseur de tension et un condensateur de filtrage. La diode permet de bloquer le courant de retour vers la batterie.

En bas du schéma se trouve le schéma de l'alimentation à découpage que nous avons récupéré sur notre schéma réalisé lors du semestre 2.

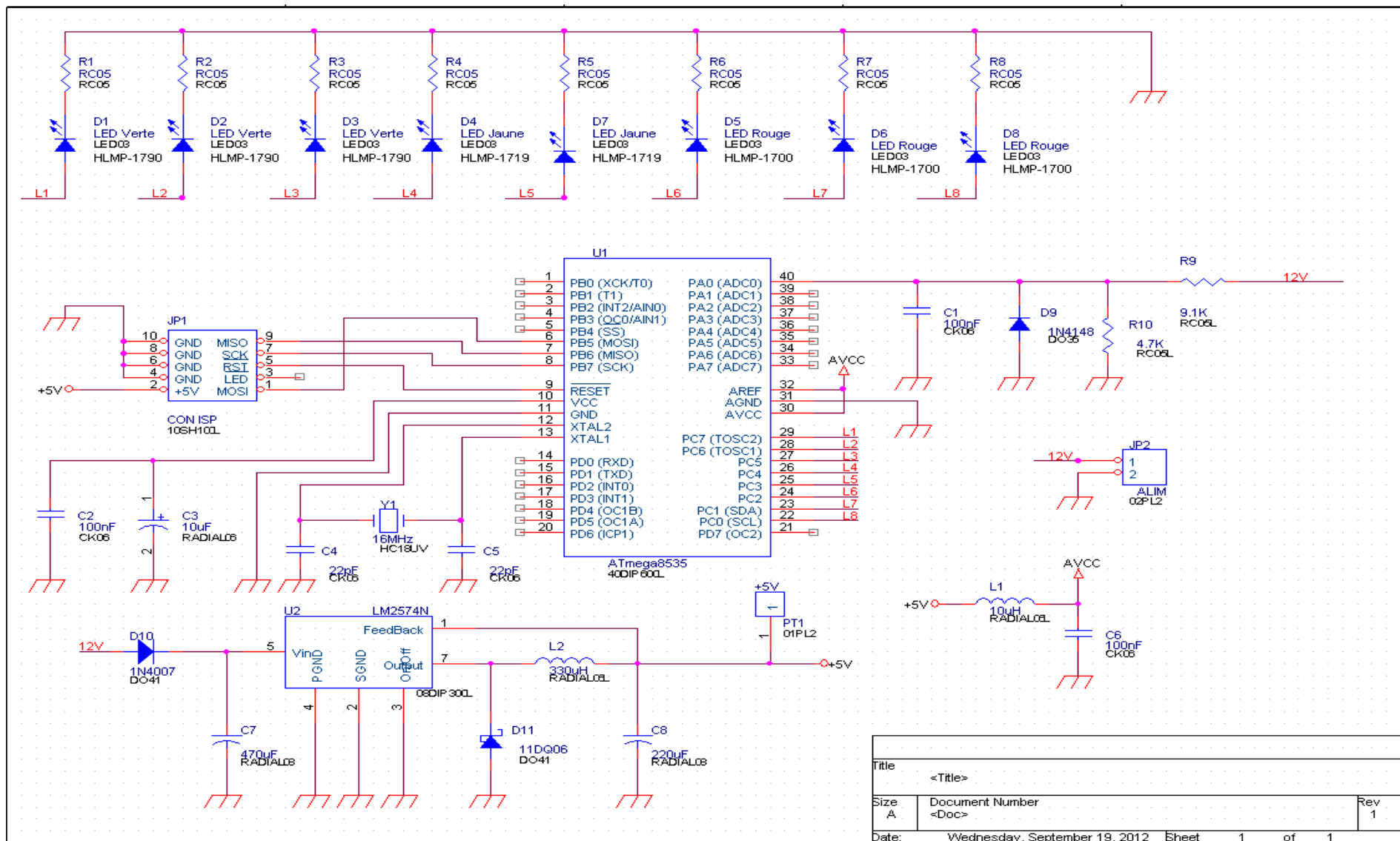


Illustration 12: Schéma structurel de la carte 1.

5.1.2. Typon, gravure et soudure

Après avoir réalisé le schéma structurel sur ORCAD nous avons placé les différents composants de manière à avoir le moins de « strape » possible. Nous avons réalisé les différentes pistes entre tous les composants de notre montage avant de réaliser le plan de masse⁷. Le rectangle jaune désigne le contour de la carte, les pistes en rouge nous montrent le cuivre et les pistes en bleu montrent les pistes que nous relirons avec des fils sur la face composants (« strape »).

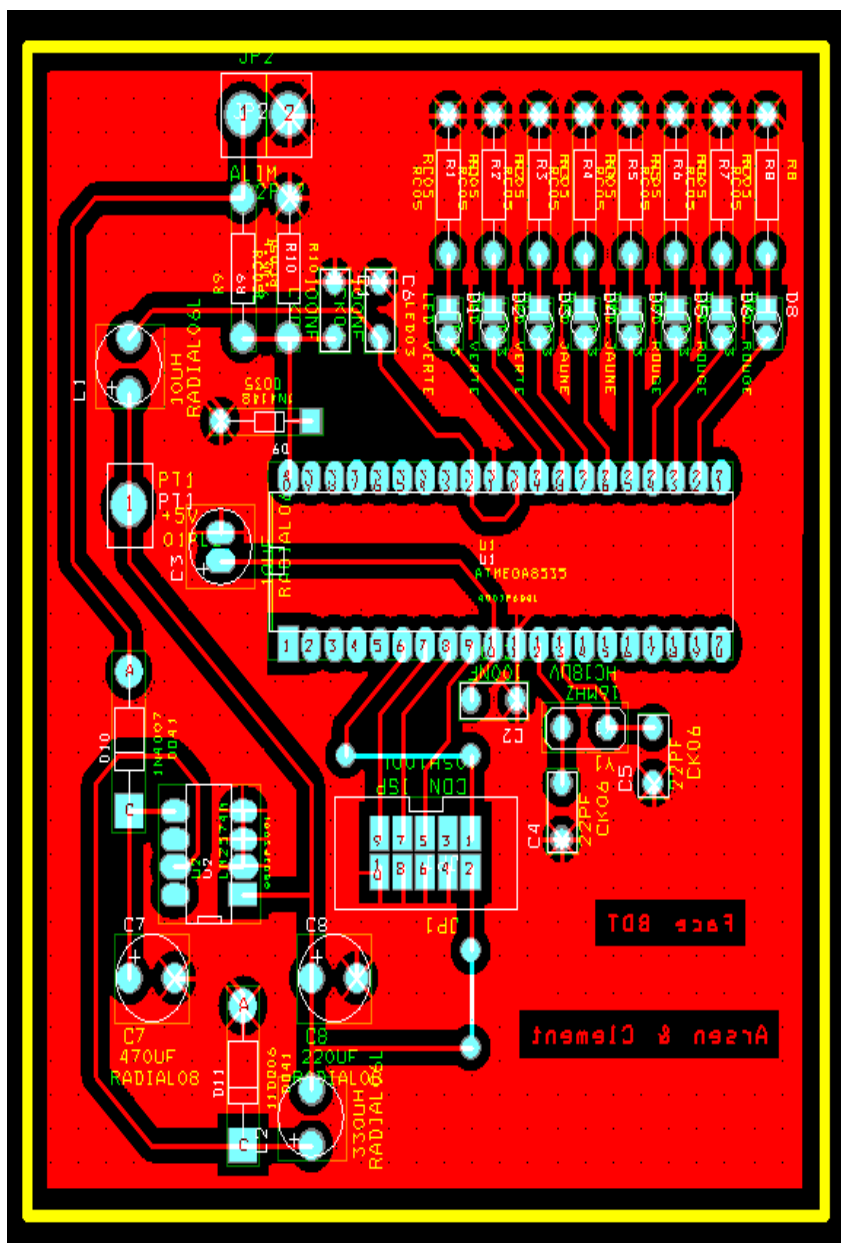


Illustration 13: Typon de la carte 1.[3]

⁷ Le plan de masse : On relie toutes les masses entre elles.

Après avoir imprimé notre typon sur un calque, nous avons insolé⁸ une carte de cuivre, après l'avoir passé au révélateur⁹ nous pouvons visualiser les pistes avec une petite couche de noir.

Ensuite nous avons gravé cette carte grâce au perchlore, cette étape nous a permis de retirer le cuivre autour des pistes.

Après ces différentes étapes nous avons effectué le perçage des différentes pastilles où nous souderons les composants par la suite. Les différents composants ont été soudés avec un fer à souder.

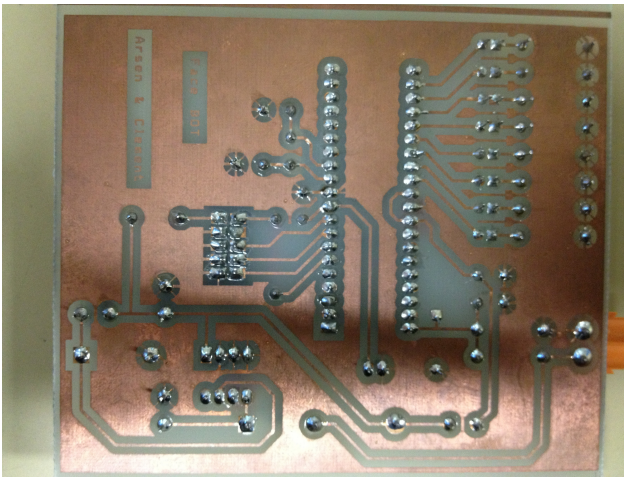


Illustration 15: Soudure de notre projet. [3]

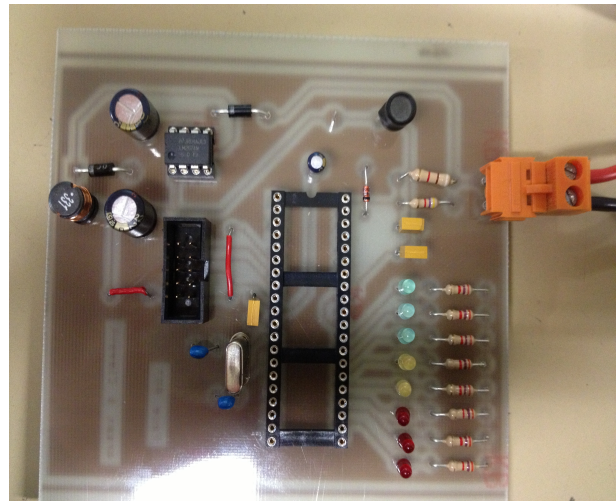


Illustration 14: Carte 1 terminé. [3]

5.1.3. Programmation de l'ATmega8535

La programmation de l'ATmega8535 a été effectuée en plusieurs étapes.

Tout d'abord, nous avons indiqué nos ports d'entrée et de sortie, un port d'entrée est laissé à 0 en hexadécimal (0x00), le port de sortie lui est mis à 255 en hexadécimal¹⁰ (0xFF).

8 Insolé : Machine permettant de marquer les pistes sur une carte de cuivre.

9 Révélateur : Produit nous permettant de faire apparaître les pistes.

10 Hexadécimal : La base hexadécimale consiste à compter sur une base 16, c'est pour cela que nous allons de 0 à F.

```

72
73 // Input/Output Ports initialization
74 // Port A initialization
75 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
76 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
77 PORTA=0x00;
78 DDRA=0x00;
79
80 // Port B initialization
81 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
82 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
83 PORTB=0x00;
84 DDRB=0x00;
85
86 // Port C initialization
87 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
88 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
89 PORTC=0x00;
90 DDRC=0xFF;
91
92 // Port D initialization
93 // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
94 // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
95 PORTD=0x00;
96 DDRD=0x00;

```

Illustration 16: Programmation des ports de sortie.

La 1^{ère} étape [1]

La 1^{ère} étape a été effectuée sur un afficheur LCD¹¹, nous avons tout d'abord vérifié si notre sortie de programmation était opérationnelle. Sur le premier ordinateur que nous utilisons cette sortie était défectueuse. Donc nous avons changé d'ordinateur et cette nouvelle sortie fonctionnait.

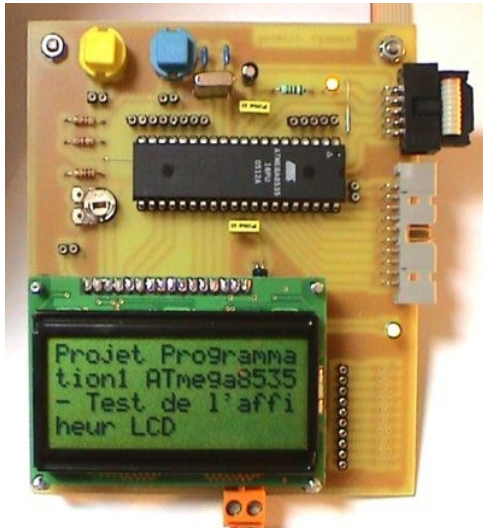
```

164
165 LCD module initialization
166 lcd_init(16);
167 lcd_gotoxy(0,0);
168 lcd_putsf("Prog Vumetre 4");
169
170 while (1)
171 {
172     TensionBat=read_adc(0);
173     a=(float)(TensionBat*14.6)/1024;
174
175     sprintf(tampon,"%4d",TensionBat);
176     lcd_gotoxy(3,1);
177     lcd_puts(tampon); //on affiche la valeur de TensionBat.
178
179     v=(float)(TensionBat*5)/1024;
180     sprintf(tampon,"Vbat = %3.3f V",v);
181     lcd_gotoxy(0,2);
182
183     lcd_puts(tampon); //on affiche la valeur de TensionBat.
184
185 }
186 };
187

```

Illustration 17: Programmation du LCD.

11 LCD :(Liquid Crystal Display), le LCD est un afficheur avec un courant de 1 à 5 mA .



Ensuite la vérification du bon fonctionnement de notre afficheur LCD a été effectuée. Son bon fonctionnement a été validé.

Nous pouvons voir le programme de test qui a été mis dans l'ATmega8535 fonctionne correctement.

Illustration 18: Validation du bon fonctionnement de l'afficheur LCD.

La 2^{ème} étape

La deuxième étape a été de visualiser la tension, donc pour cela nous devons faire une Conversion Analogique Numérique (CAN).

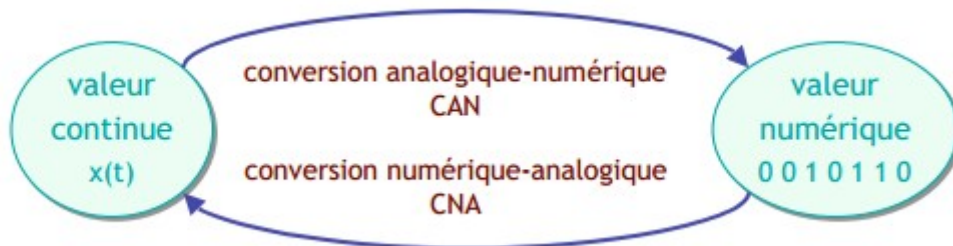


Illustration 19: Conversion Analogique Numérique.[4]

La tension que l'on affichait était une tension entière, nous voulions une tension décimale donc pour cela nous créons une variable « float V ; ». Nous créons aussi une variable « float TensionBat ; ».

$$0V \rightarrow 5V$$

$$0 \rightarrow 1024$$

La formule que nous avons trouvée pour nous permettre d'afficher la tension de la batterie sur l'afficheur LCD est $V = (float) \left(\frac{TensionBat \times 5}{1024} \right)$

La valeur maximale que nous visualiserons sera de 14,5V donc on remplace le 5 par 14,5.

$$V = (float) \left(\frac{TensionBat \times 14,5}{1024} \right)$$

Cette formule nous permettra d'afficher sur l'écran LCD la tension entre 10V et 14,5V.

La 3ème étape

Nous avons créé le programme pour que le micro-contrôleur allume des LED au lieu d'afficher la tension sur l'écran LCD.

Afin que le programme puisse fonctionner, il a fallu retirer tout les lignes de programme que nous avions d'écrits à l'étape précédente. Par la suite, nous avons effectué le test sur une carte à LED.

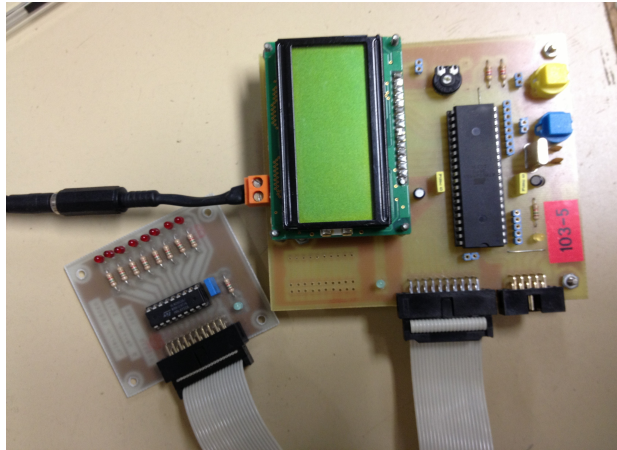


Illustration 20: Teste avec une carte de programmation et des LED.

Nous avons réalisé la programmation de l'ATmega8535 pour vérifier si notre programme fonctionnait.

La déclaration des variables sur les ports de sortie se fait avec des « #define » comme nous pouvons le voir dans le 1^{er} cadre. Dans le deuxième cadre nous avons les déclarations de nos autres variables qui sont utilisées dans le programme.

Sur le site de M.LEQUEU [1] nous avons trouvé des informations qui nous ont permis de programmer des variables sur les différents ports de sortie.

```

38 #define ADC_VREF_TYPE 0x00
39
40 #define S7 PORTC.7
41 #define S6 PORTC.6
42 #define S5 PORTC.5
43 #define S4 PORTC.4
44 #define S3 PORTC.3
45 #define S2 PORTC.2
46 #define S1 PORTC.1
47 #define S0 PORTC.0
48
49 // Read the AD conversion result
50 unsigned int read_adc(unsigned char adc_input)
51 {
52     ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
53     // Delay needed for the stabilization of the ADC input voltage
54     delay_us(10);
55     // Start the AD conversion
56     ADCSRA|=0x40;
57     // Wait for the AD conversion to complete
58     while ((ADCSRA & 0x10)==0);
59     ADCSRA|=0x10;
60     return ADCW;
61 }
62
63 // Declare your global variables here
64 unsigned int TensionBat;
65 float a=10;
66
67

```

Illustration 21: Déclaration des variables.[3]

Ordinogramme [3]

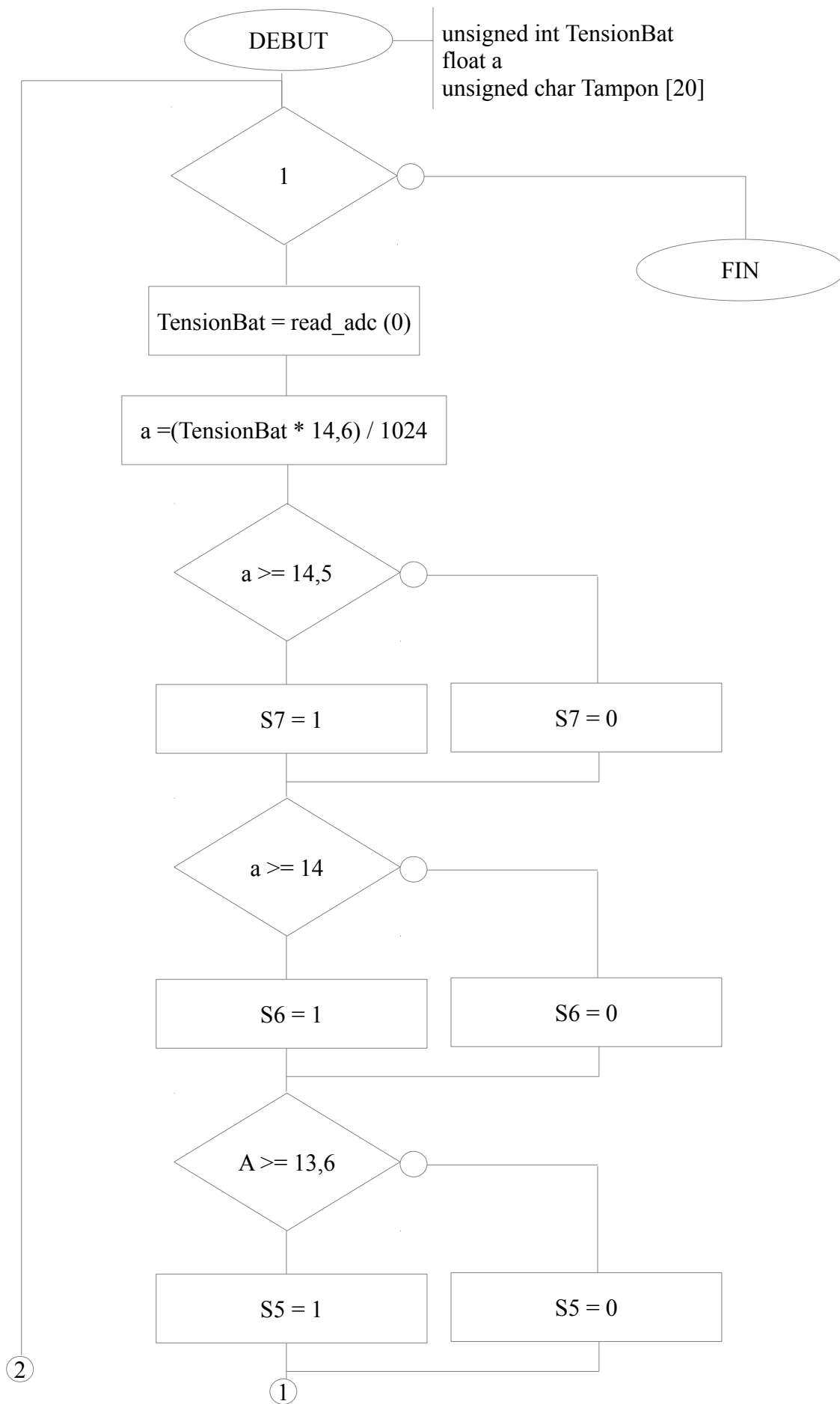
Avant d'écrire le code sur le logiciel code vision CVR nous avons effectué l'ordinogramme suivant.

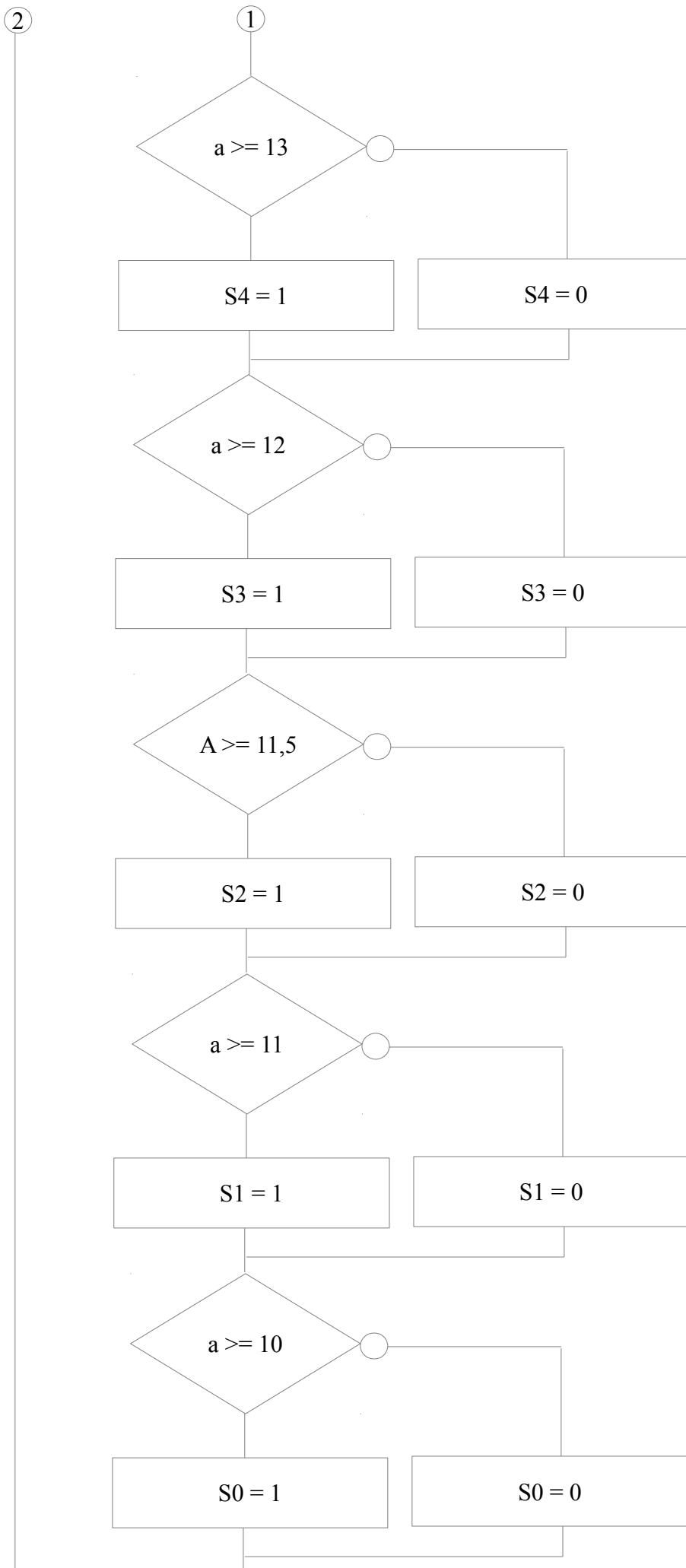
Nous voulions que le micro-contrôleur lise la valeur de la batterie donc nous utilisons la fonction `TensionBat = read_adc (0)`, à l'étape précédente nous avons trouvé la formule qui permet au micro-contrôleur la conversion de la tension d'entrée qui est à plus ou moins 5V en une tension entre 10V et 14,5V, car il ne peut y avoir en entrée une tension supérieure à 5V (rôle du pont diviseur expliqué précédemment). Ce qui nous permettra d'écrire dans le programme les tensions comprises entre 10 et 14,5V.

$$a = (\text{float}) \left(\frac{\text{TensionBat} \times 14,5}{1024} \right)$$

Ensuite nous testons si la valeur de la batterie est supérieure à 14,5V la LED 8 s'allume sinon il faut qu'elle reste éteinte. Nous effectuons ce test sur les tensions 14V (LED 7), 13,6V (LED 6), 13V (LED 5), 12V (LED 4), 11,5V (LED 3), 11V (LED 2) et 10V (LED 1).

Pour allumer une LED il faut que la variable de son port soit à 1, par exemple pour allumer la LED 8 il faut que $S7 = 1$. Alors que pour qu'elle reste éteinte il faut que le port de sortie reste à 0 ($S7 = 0$).





Codage de l'ordinogramme [3]

```
170 while (1)
171 {
172     TensionBat=read_adc(0);
173     a=(float)(TensionBat*14.6)/1024;
174
175
176     if(a>=14.5)
177     {
178         S7=1;
179     }
180     else
181     {
182         S7=0;
183     };
184
185     if(a>=14)
186     {
187         S6=1;
188     }
189     else
190     {
191         S6=0;
192     }
193     if(a>=13.6 )
194     {
195         S5=1;
196     }
197     else
198     {
199         S5=0;
200     }
201     if(a>=13)
202     {
203         S4=1;
204     }
205     else
206     {
207         S4=0;
208     }
209     if(a>=12)
210     {
211         S3=1;
212     }
213     else
214     {
215         S3=0;
216     }
217 }
```

Illustration 22: Code du Vu Mètre Partie 1.

Dans le premier cadre nous retrouvons bien la fonction et la formule de notre ordinogramme.

Et après, les différents tests qui seront effectués pour connaître la tension de la batterie.


```
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
};  
  
if(a>=11.5)  
{  
    S2=1;  
}  
else  
{  
    S2=0;  
}  
if(a>=11)  
{  
    S1=1;  
}  
else  
{  
    S1=0;  
}  
if(a>=10)  
{  
    S0=1;  
}  
else  
{  
    S0=0;  
}  
}
```

Illustration 23: Code du Vu Mètre Partie 2.

Après avoir écrit tout le programme nous l'avons transféré dans l'ATmega8535 afin de le tester sur notre carte de test. Nous avons ensuite validé son bon fonctionnement à l'aide d'une alimentation variable.

La 4ème étape

Dans cette étape nous avons transféré dans notre micro-contrôleur le programme afin de valider le bon fonctionnement de notre carte. Après avoir validé son fonctionnement nous avons recherché une boîte pour la protéger. Durant cette recherche nous avons trouvé différentes boîtes, donc nous avons choisi avec notre professeur de refaire une carte plus petite pour qu'elle soit plus pratique.

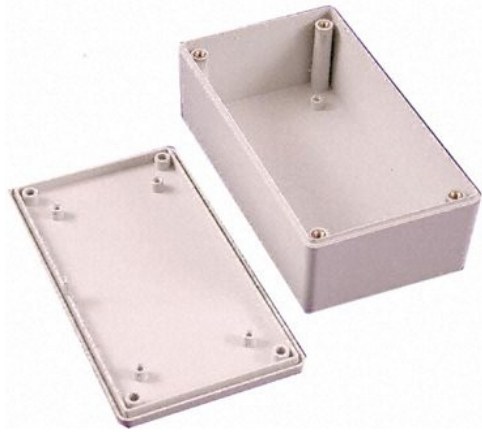


Illustration 24: Boite de la carte 1.

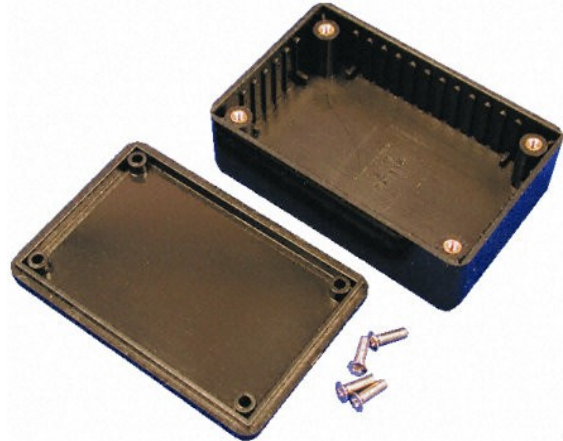


Illustration 25: Boite de la carte 2.

5.2. Réalisation de la carte 2

Pour la réalisation de cette nouvelle carte nous avons choisi la boite avant de faire la carte. Le schéma structurel et le typon ont été modifiés, car l'ATmega8535 que nous avons utilisé n'a pas la même forme. C'est un ATmega8535 en carré de 44 broches. Seul le boîtier de l'ATmega8535 change.

5.2.1. Schéma structurel

Voici le nouveau schéma structurel de notre carte. Les numéros de broches de l'ATmega8535 utilisés pour la carte 2 ont été modifié. En annexe 2 vous pourrez voir le schéma de notre nouvelle ATmega8535, cette annexe nous a permis de réaliser sur ORCAD le schéma de notre ATmega8535.

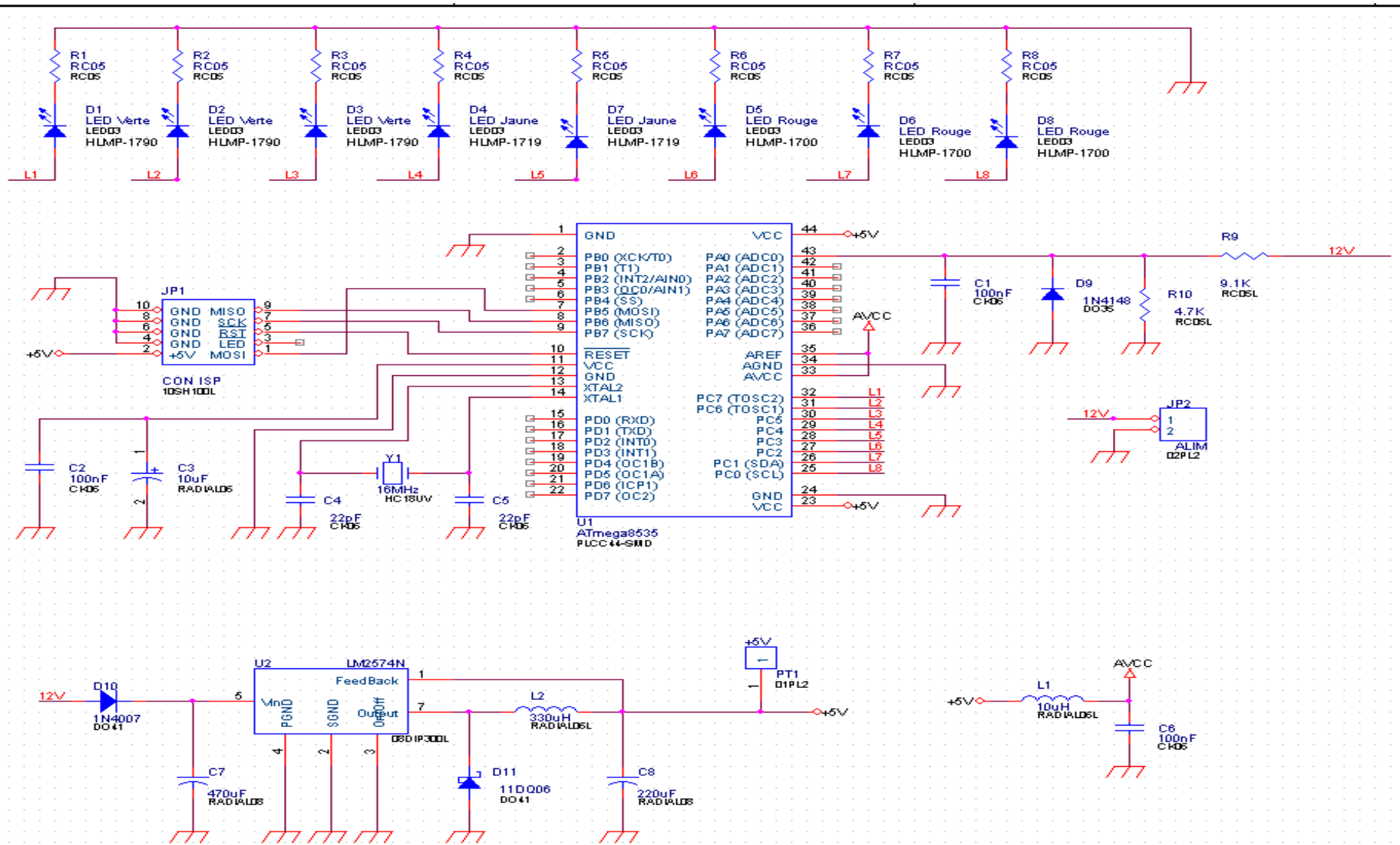


Illustration 26: Schéma structurel de la carte 2. [3]

5.2.2. Typon, gravure et soudure

Nous avons modifié le contour de la carte pour que celle-ci puisse tenir dans la boîte. Ensuite on a placé les composants et tracé leurs différentes pistes.

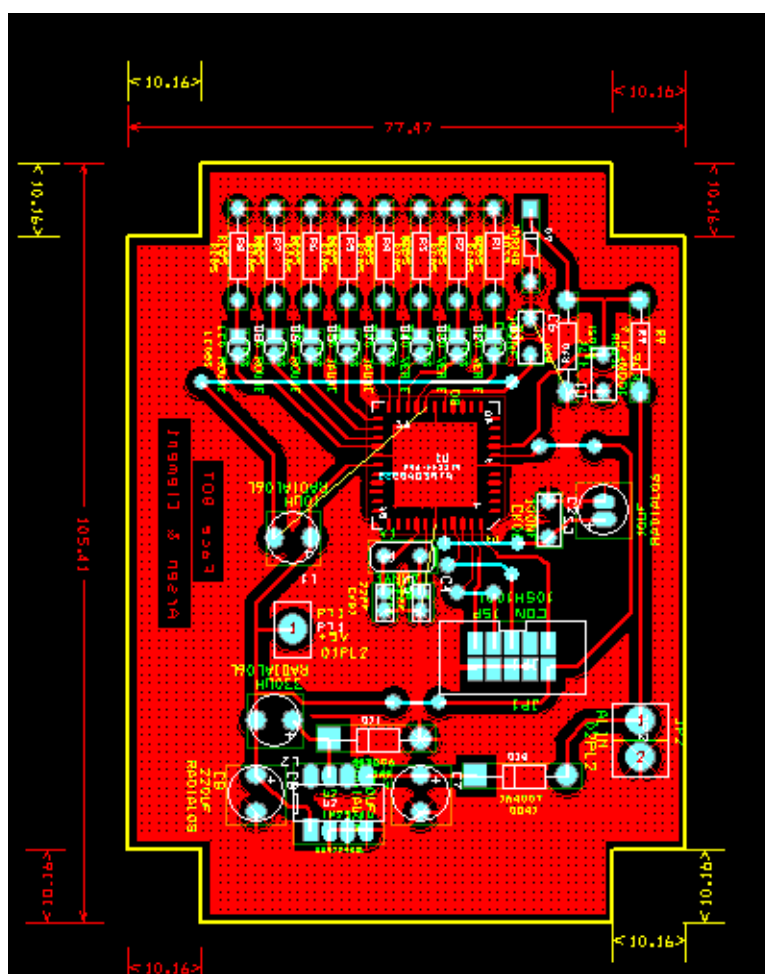


Illustration 27: Typon de la carte 2.

La gravure reste identique à la carte précédente.

La soudure

Pour la soudure nous avons été obligés d'utiliser une soudure cms¹². Celle ci est différente de la soudure classique (au fer à souder). Pour la soudure cms nous utilisons de la patte à souder que nous étalons sur les pastilles de la carte. Par la suite on pose les broches des composants sur cette patte à souder et nous passons la carte au four (environ 2,30 minutes à 170° et 30 secondes à 300°).

On voit sur l'image suivante la soudure cms de l'ATmega8535 sur la carte.

Sur la deuxième image nous pouvons voir les différentes broches du micro-contrôleur soudés.

¹² Les composants sont des composants montés en surface, ils n'ont pas besoin de perçage.

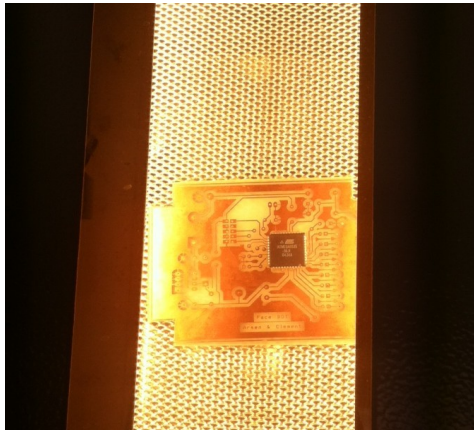


Illustration 29: Carte dans le four.

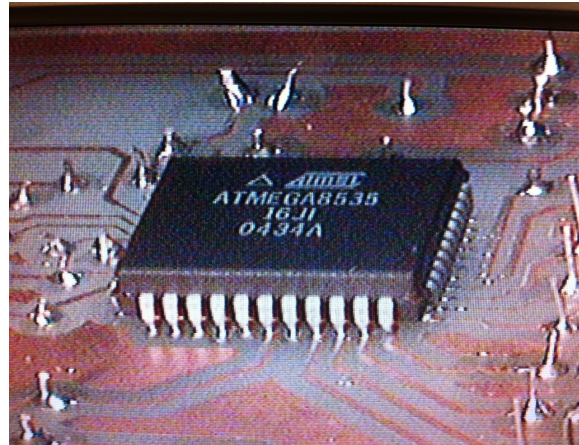


Illustration 28: Vérification des différentes soudures.

5.2.3. Programmation de l'ATmega8535

Le programme que l'on utilise pour la deuxième carte est identique à la première, le port de sortie reste le port C.

5.3. Problèmes rencontrés

1^{er} problème

Après avoir programmé la carte 2, on a remarqué que durant le test une des LED jaune ne s'allumait pas. Nous nous sommes posés la question si cette LED avait un dysfonctionnement, nous l'avons donc remplacé. Après avoir testé une deuxième fois cette LED ne fonctionnait toujours pas.

Nous avons vérifié la tension délivrée par le micro-contrôleur sur les différentes LED, on a pu alors remarquer que le micro-contrôleur ne délivrait pas de tension sur cette LED jaune, car cette broche n'était pas soudée.



Illustration 30: Patte de l'ATmega8535 non soudée.

Résolution du problème

Nous avons soudé cette patte à l'aide d'un fer à souder et d'étain. Ensuite notre carte 2 fonctionnait très bien.

2^{ème} problème

Lors de la mise en boîte de la carte 2, la carte ne rentrait pas dedans, car elle était trop grande. Nous n'avions pas pris en compte les petites lamelles situées à l'intérieur de la boîte.

Voir annexe 1.

Résolution du problème

Nous avons dû couper les bords de notre carte. Ensuite nous avons pu l'insérer dans notre boîte.

5.4. Nomenclature

Voici la nomenclature de notre projet. [3]

Nomenclature des cartes							
Repère	Désignation	Type	Caractéristique				
			Valeur	Puissance Tension	Nombre	Prix unitaire	Prix total
Alimentation micro-contrôleur							
Y1	16MHz				1	0,430 €	0,430 €
C1/C2/C6	condensateurs 100nF			63V	3	0,104 €	0,312 €
C3	Condensateur 10µF				1	1,160 €	1,160 €
C4/C5	Condensateurs 22pF				2	0,174 €	0,348 €
C7	Condensateur 470µF				1	0,140 €	0,140 €
C8	Condensateur 220µF				1	3,360 €	3,360 €
D9	Diode de redressement	1N4148			1	0,011 €	0,011 €
D10	Diode de redressement	1N4007			1	0,190 €	0,190 €
D11	Diode Schottky	11DQ06		1,1A, 60V	1	0,190 €	0,190 €
L1	Inductance		470uH	0,5A		0,100 €	0,000 €
LM2574	Régulateur de tension	LM 2574		5V	1	1,190 €	1,190 €
Alimentation des LED							
R1 à R8	Résistances de protection des LED		1,5KΩ		8	0,142 €	1,136 €
D1 à D3	LED Verte (Diode électroluminescente)			2mA	3	0,168 €	0,504 €
D4 et D5	LED Jaune (Diode électroluminescente)			2mA	2	0,168 €	0,336 €
D6 à D8	LED Rouge (Diode électroluminescente)			2mA	3	0,168 €	0,504 €
Micro-contrôleur							
U1	Micro-contrôleur	ATM8535			1	3,750 €	3,750 €
Autre							
Boîtier	Boîtier ABS Noir IP54 110*82*40				1	5,500 €	5,500 €
	Presse étoupe				1	9,320 €	9,320 €
Total des dépenses							28,381 €

Conclusion

Dans ce dossier, on a établi un cahier des charges où nous avons présenté le sujet et ses contraintes.

Ensuite, une analyse générale de notre projet a été faite pour définir les différentes parties qui seront étudiées dans l'analyse technique. Celle-ci est composée de trois parties, alimentation à découpage, réduction de la tension, contrôle des LED suivant la tension d'entrée.

Nos différentes cartes que nous avons réalisées sont expliquées dans la cinquième partie. Nous vous expliquons les différentes étapes pour la réalisation de nos cartes. Notre carte 1 nous a permis de vérifier le bon fonctionnement du schéma structurel et celui de notre programme. Après avoir validé son bon fonctionnement nous avons décidé de refaire une carte plus petite. On vous a présenté aussi les problèmes que nous avons eus sur la deuxième carte. Une de nos LED ne fonctionnait pas donc nous avons recherché d'où venait problème et nous l'avons corrigé.

Cette expérience nous a permis d'utiliser différents montages que nous avons étudiés. Ces montages ainsi insérés dans un projet rajoutent une dimension concrète au travail effectué les semestres passés. Nous avons aussi appris à travailler en équipe.

Résumé

Ce dossier présente un Vu-Mètre pour batterie de voiture 12 V pour cela il est constitué de cinq parties.

Dans une première partie, le cahier des charges a été expliqué. Dans celui-ci, la présentation du projet, les fonctions, les contraintes et les caractéristiques y sont exposées.

La deuxième partie est constituée du planning prévisionnel et du planning réel. Les différences entre ces deux plannings sont présentées.

L'analyse générale du projet constitue une troisième partie. Elle relate rapidement la fonction principale du projet avec des schémas fonctionnels (niveau 1 et niveau 2) et le sépare donc en plusieurs parties techniques ces différentes parties seront expliquées dans l'analyse technique.

Dans une quatrième partie, l'analyse technique est faite. Celle-ci est séparée en trois grandes parties qui expliquent les solutions techniques du projet. La première partie est l'alimentation à découpage qui nous sert à alimenter le micro-contrôleur. La deuxième partie est la réduction de tension qui est faite grâce à un pont diviseur de tension. Dans la troisième partie nous expliquons le contrôle des LED suivant la tension d'entrée, à l'aide du micro-contrôleur on contrôle l'allumage des LED en fonction de la tension d'entrée.

La partie réalisation expose les différentes cartes réalisées ainsi que les différents problèmes rencontrés avec notre deuxième carte. Nous expliquons la programmation de notre ATmega8535. Dans la dernière sous partie nous vous montrons coût total de notre Vu-Mètre.

(228 mots)

Bibliographie

[1] **Thierry LEQUEU**. *La documentation de Thierry LEQUEU sur OVH*, 22 octobre 2012, [En ligne]. (Page consultée le 24/10/2012) <<http://www.thierry-lequeu.fr/>>

[2] **Nina**. *Montage pont diviseur de tension*, 05/06/2012, [En ligne]. (Page consultée le 05/10/2012) <<http://www.astuces-pratiques.fr/electronique/le-pont-diviseur-de-tension>>

[3] **ARAKELIAN et GALLIER**. *"Production personnelle"*. ARAKELIAN et GALLIER, 2012.

[4] **CNRS Orléans**. *"Conversion analogique numérique"*, , [En ligne]. (Page consultée le 24/10/2012) <http://lpc2e.cnrs-orleans.fr/~ddwit/enseignement/cours_elec_num_5.pdf>

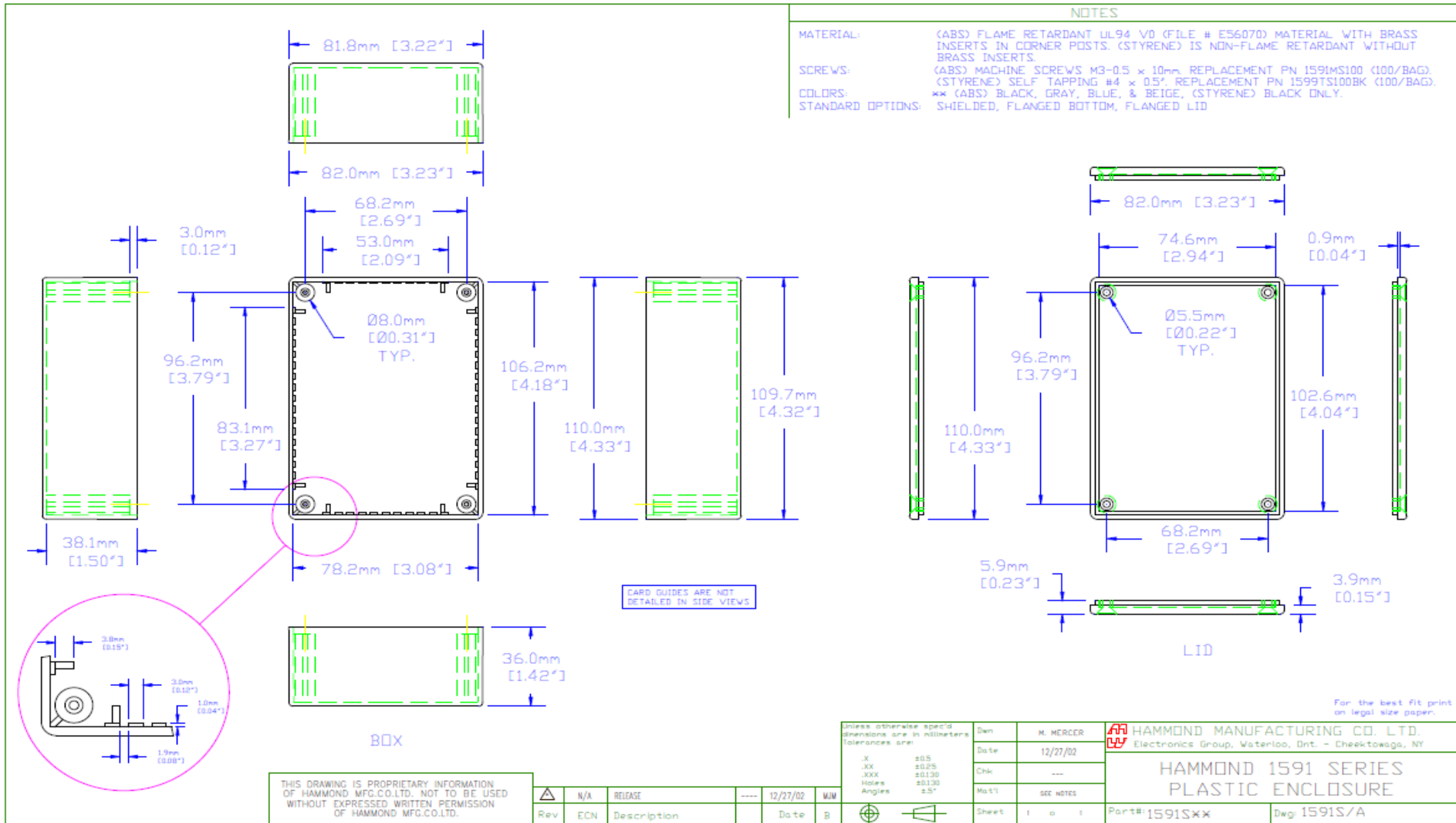
Index des illustrations

Illustration 1: Planning.....	7
Illustration 2: Schéma fonctionnel de niveau 1 du projet. [3].....	8
Illustration 3: Schéma fonctionnel de niveau 2 du projet. [3].....	9
Illustration 4: Schéma fonctionnel de niveau 1 de l'alimentation à découpage. [3].....	9
Illustration 5: Schéma structurel d'un réducteur de tension. [1].....	10
Illustration 6: Schéma fonctionnel de niveau 1 de la réduction de tension. [3].....	11
Illustration 7: Pont diviseur de tension. [2].....	11
Illustration 8: Schéma fonctionnel du Contrôle des LED.[3].....	12
Illustration 9: Icône de code vision AVR.....	12
Illustration 10: ATmega8535 en longueur de 40 pattes. [1].....	13
Illustration 11: ATmega8535 en carré de 44 pattes.....	13
Illustration 12: Schéma structurel de la carte 1.....	15
Illustration 13: Typon de la carte 1.[3].....	16
Illustration 14: Carte 1 terminé. [3].....	17
Illustration 15: Soudure de notre projet. [3].....	17
Illustration 16: Programmation des ports de sortie.....	18
Illustration 18: Programmation du LCD.....	18
Illustration 17: Validation du bon fonctionnement de l'afficheur LCD.....	19
Illustration 19: Conversion Analogique Numérique.[4].....	19
Illustration 20: Teste avec une carte de programmation et des LED.....	20
Illustration 21: Déclaration des variables.[3].....	21
Illustration 22: Code du Vu Mètre Partie 1.....	24
Illustration 23: Code du Vu Mètre Partie 2.....	25
Illustration 24: Boîte de la carte 1.....	26
Illustration 25: Boîte de la carte 2.....	26
Illustration 26: Schéma structurel de la carte 2. [3].....	27
Illustration 27: Typon de la carte 2.....	28
Illustration 28: Vérification des différentes soudure.....	29
Illustration 29: Carte dans le four.....	29
Illustration 30: Patte de l'ATmega8535 non soudé.....	29

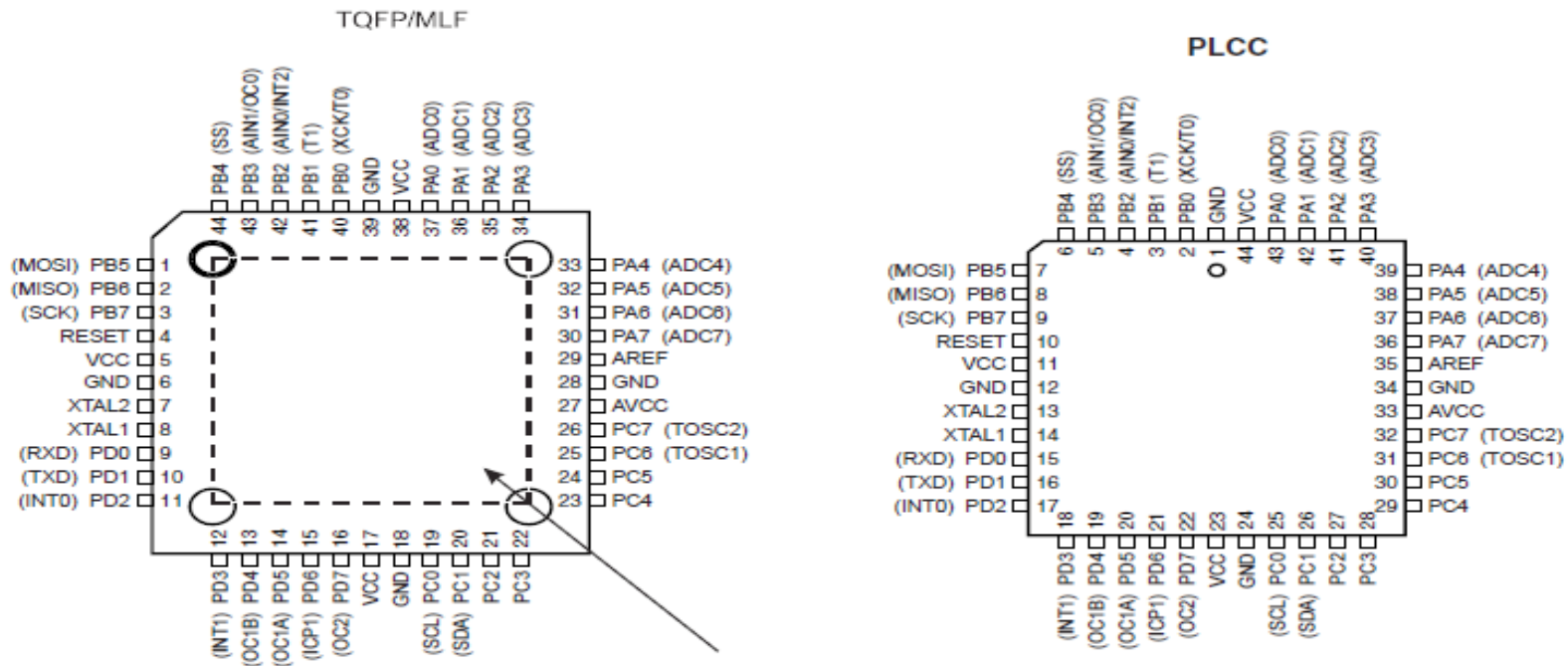
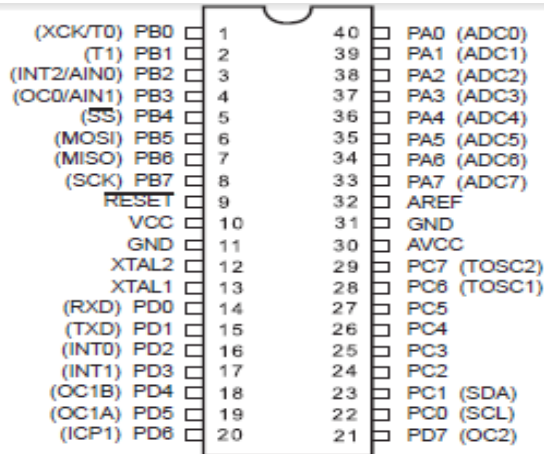
Index des mots clé

LED (light-emitting diode), c'est des diodes électroluminescente.....	8
5V DC signifie 5V courant continu.....	9
LM2574 : c'est un régulateur de tension (12 V à 5 V DC).....	10
ATmega8535 : c'est le composant programmable qui permet de gérer la commande de coupure de l'alimentation (tension de 5 V DC).....	12
Vcc est la tension d'alimentation principale de l'ATmega.....	12
Loi d'ohm : $U = R \times I$	13
Le plan de masse : On relie toutes les masses entre elles.....	16
Insolé : Machine permettant de marquer les pistes sur une carte de cuivre.....	17
Révélateur : Produit nous permettant de faire apparaître les pistes.....	17
Hexadécimal : La base hexadécimale consiste à compter sur une base 16, c'est pour cela que nous allons de 0 à F.....	17
LCD :(Liquid Crystal Display), le LCD est un afficheur avec un courant de 1 à 5 mA	18
Les composants sont des composants montés en surface, ils n'ont pas besoin de perçage.....	28

Annexe 1 : Schéma de construction de la boîte pour la carte 2



Annexe 2 : Schéma du 2ème ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

[1][1]

[2]

[3][3][3]

[4]